CSE176 Introduction to Machine Learning — Homework set #2

Fall semester 2023 — Miguel Á. Carreira-Perpiñán

Exercise 1: bias and variance of an estimator (12 points). Assume we have a sample $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}$ of N iid (independent identically distributed) scalar random variables, each of which is drawn from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu \in \mathbb{R}$ and $\sigma > 0$. We want to estimate the mean μ of this Gaussian by computing a statistic of the sample $\mathcal{X}$. Consider the following four different statistics of the sample:

1. $\phi_1(\mathcal{X}) = 7$.
2. $\phi_2(\mathcal{X}) = x_1$.
3. $\phi_3(\mathcal{X}) = \frac{1}{N} \sum_{n=1}^N x_n$.
4. $\phi_4(\mathcal{X}) = x_1 x_2$.

For each statistic ϕ, compute:

- (1 point) Its bias $b_\mu(\phi) = \mathrm{E}_{\mathcal{X}}\left\{\phi(\mathcal{X})\right\} - \mu$.
- (1 point) Its variance $\mathrm{var}\left\{\phi\right\} = \mathrm{E}_{\mathcal{X}}\left\{(\phi(\mathcal{X}) - \mathrm{E}_{\mathcal{X}}\left\{\phi(\mathcal{X})\right\})^2\right\}$.
- (1 point) Its mean square error $e(\phi, \mu) = \mathrm{E}_{\mathcal{X}}\left\{(\phi(\mathcal{X}) - \mu)^2\right\}$.

Based on that, answer the following for each estimator (statistic): is it unbiased? is it consistent?

Hint: expectations wrt the distribution of the N-point sample $\mathcal{X}$ are like this one:

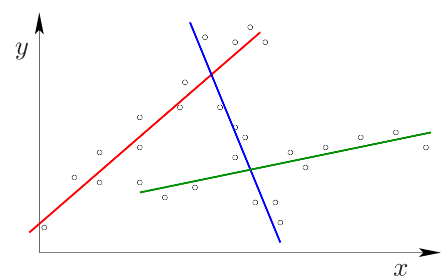
$$\mathrm{E}_{\mathcal{X}}\left\{\phi(\mathcal{X})\right\} = \int \phi(\mathbf{x}_1, \dots, \mathbf{x}_N)\, p(\mathbf{x}_1, \dots, \mathbf{x}_N)\, d\mathbf{x}_1 \dots d\mathbf{x}_N \overset{\text{iid}}{=} \int \phi(\mathbf{x}_1, \dots, \mathbf{x}_N)\, p(\mathbf{x}_1) \dots p(\mathbf{x}_N)\, d\mathbf{x}_1 \dots d\mathbf{x}_N.$$

Exercise 2: variation of k-means clustering (18 points). Consider the k-means error function:

$$E(\{\boldsymbol{\mu}_k\}_{k=1}^K, \mathbf{Z}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \left\|\mathbf{x}_n - \boldsymbol{\mu}_k\right\|^2 \quad \text{s.t.} \quad \mathbf{Z} \in \{0,1\}^{NK},\ \mathbf{Z}\mathbf{1} = \mathbf{1}$$

over the centroids $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K$ and cluster assignments $\mathbf{Z}_{N \times K}$, given training points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^D$.

- **Variation:** in k-means, we seek K clusters, each characterized by a centroid $\boldsymbol{\mu}_k$. Imagine we seek instead K lines (or hyperplanes, in general), each characterized by a weight vector $\mathbf{w}_k \in \mathbb{R}^D$ and bias $w_{k0} \in \mathbb{R}$, given a supervised dataset $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ (see figure). Data points assigned to line k should have minimum least-squares error $\sum_{n \in \text{line } k} (y_n - \mathbf{w}_k^T \mathbf{x}_n - w_{k0})^2$.
  1. (7 points) Give an error function that allows us to learn the lines' parameters $\{\mathbf{w}_k, w_{k0}\}_{k=1}^K$.
  2. (11 points) Give an iterative algorithm that minimizes that error function.



Exercise 3: PCA and LDA (30 points). Consider 2D data points coming from a mixture of two Gaussians with equal proportions, different means, and equal, diagonal covariances (where $\mu, \sigma_1, \sigma_2 > 0$):

$$\mathbf{x} \in \mathbb{R}^2\colon\ p(\mathbf{x}) = \pi_1\, p(\mathbf{x}|1) + \pi_2\, p(\mathbf{x}|2) \qquad p(\mathbf{x}|1) \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \quad p(\mathbf{x}|2) \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2),$$

$$\pi_1 = \pi_2 = \frac{1}{2}, \quad \boldsymbol{\mu}_1 = \mathbf{0},\ \boldsymbol{\mu}_2 = \begin{pmatrix} \mu \\ 0 \end{pmatrix}, \quad \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}.$$

1. (5 points) Compute the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of the mixture distribution $p(\mathbf{x})$.

   Hint: let $p(\mathbf{x}) = \sum_{k=1}^K \pi_k\, p(\mathbf{x}|k)$ for $\mathbf{x} \in \mathbb{R}^D$ be a mixture of K densities, where $\pi_1, \dots, \pi_K \in [0,1]$ and $\sum_{k=1}^K \pi_k = 1$ are the component proportions (prior probabilities) and $p(\mathbf{x}|k)$, for $k = 1, \dots, K$, the component densities (e.g. Gaussian, but not necessarily). Let $\boldsymbol{\mu}_k = \mathrm{E}_{p(\mathbf{x}|k)}\left\{\mathbf{x}\right\}$ and $\boldsymbol{\Sigma}_k = \mathrm{E}_{p(\mathbf{x}|k)}\left\{(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T\right\}$ be the mean and covariance of component density k, for $k = 1, \dots, K$. Then, the mean and covariance of the mixture are (you should be able to prove this statement):

$$\boldsymbol{\mu} = \mathrm{E}_{p(\mathbf{x})}\left\{\mathbf{x}\right\} = \sum_{k=1}^K \pi_k \boldsymbol{\mu}_k \qquad \boldsymbol{\Sigma} = \mathrm{E}_{p(\mathbf{x})}\left\{(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\right\} = \sum_{k=1}^K \pi_k \left(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T\right) - \boldsymbol{\mu}\boldsymbol{\mu}^T.$$

2. (5 points) Compute the eigenvalues $\lambda_1 \geq \lambda_2 \geq 0$ and corresponding eigenvectors $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^2$ of $\mathbf{\Sigma}$. Can we have $\lambda_2 > 0$?
3. (2 points) Find the PCA projection to dimension 1.
4. (5 points) Compute the within-class and between-class scatter matrices $\mathbf{S}_W$, $\mathbf{S}_B$ of p.
5. (6 points) Compute the eigenvalues $\nu_1 \geq \nu_2 \geq 0$ and corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^2$ of $\mathbf{S}_W^{-1}\mathbf{S}_B$. Can we have $\nu_2 > 0$?
6. (2 points) Compute the LDA projection.
7. (5 points) When does PCA find the same projection as LDA? Give a condition and explain it.

Exercise 4: construct example classifiers in 2D (40 points). Consider the 2D dataset in the figure, having $K = 3$ classes (red, green and blue), and which is split into training (+ markers, $N = 16$ instances) and validation ($\circ$ markers, 8 instances). You are going to construct several classifiers *by hand*, as best as you can (do not run any software on the data) and answer several questions. To construct each classifier, use *only* the training set.

k-nearest-neighbor classifier

- (4 points) The Voronoi cell for an instance $\mathbf{x}_n \in \mathbb{R}^2$ is the set of input space points closer to $\mathbf{x}_n$ (in Euclidean distance) than to any other instance. The Voronoi tessellation is a partition of the space into such cells. Sketch this tessellation and explain how you do it.
- (2 points) Using $k = 1$, construct a k-nearest-neighbor classifier. On the figure, indicate the region for each class and the class boundaries. Explain how you do it.
- (2 points) Apply this classifier to every training instance and give the training error in 0/1 loss (number of misclassified instances). Do the same for the validation set.

Classification tree

- (7 points) Using the following algorithm *exactly as stated*, construct a classification tree. The tree is binary (two children at each decision node), axis-aligned (testing a single feature/threshold at each decision node, e.g. "$x_2 \geq 3.5$") and each leaf predicts a single class label. The algorithm is a variation of greedy recursive partitioning where we select the feature in a cyclic order (rather than by optimizing purity), we take the threshold to be the median of the values, and we continue to grow the tree until it reaches a depth of 3 (thus having $2^3 = 8$ leaves). Call the threshold values as $a_1, a_2, a_3 \dots$ and $b_1, b_2, b_3 \dots$ for the features x_1 and x_2, respectively. Specifically:
  - Depth 0. Start with a root node. Using feature x_1, determine its threshold a_1 (median). Its decision is "if $x_1 \geq a_1$ go right, else go left".
  - Depth 1. For each of its children, do the same but using feature x_2, and thresholds b_1 and b_2 for each of the children.
  - Depth 2. Repeat but using feature x_1.
  - Depth 3. We are at the leaves. Label each with the majority class. Break ties by this order of preference: red > green > blue.

  Draw the tree itself, indicating at each decision node its feature/threshold and at each leaf its label. On the figure, indicate the region for each class and the class boundaries.
- (2 points) Apply this classifier to every training instance and give the training error in 0/1 loss (number of misclassified instances). Do the same for the validation set.
- (2 points) Extract IF-THEN rules from the tree.
- (5 points) Repeat everything but with a second tree, constructed with the same algorithm but starting at the root with feature x_2 and then cycling over features as before.
- (5 points) Based on these results, suggest ways in which the tree algorithm above could be improved. The more insightful your suggestions, the better the grade.

Classification forest

- (2 points) Consider the two trees above but grown up to depth 2 only, and label accordingly the 4 leaves.

- (7 points) Ensemble these two trees into a forest, whose predicted label is the majority vote over the two trees. Break ties as before, by this order of preference: red > green > blue.
  On the figure, indicate the region for each class and the class boundaries. Make sure to identify each region of the space created by the forest, and the label for that region.